Issue Number 5 — September 1983

STOP PRESS DRAGON

DRAGON RELEASES DISKS

Dragon Data Ltd has launched a Disk Drive unit for the Dragon 32 home computer

The Disk Drive will expand the power of your 32 speed up program loading and data file handling It is the next step up in data and program storage from the cassette recorder and can itself be expanded as you and your Dragon system progress Priced at £275, the Dragon Disk Drive is a single half height drive in a coated steel case tt has an internel power supply end is easily expandable to a double disk system by inserting an additional drive Two double units can be linked to form a 4-drive system

its specifications are

Disk Typa
5¼" Mini diskette
Mamory Cepecity
(Formated) 184320 bytes
Disk Organisation
Single sided
Double density
40 tracks (TPI)
18 sectors per track
256 bytes per sector
Directory on track 20
Case
Coated steel, capable of holding two half height drives
Weight (with ona driva)
4 4Kg

The controller can support up to four drives, single or double sided capability Up to ten files may be open simultaneously The disk operating system is held in ROM (Read memory only) on the controller card

The Dragon Disk Drives will be available through the usuel Dragon dealerships and retailers, including Boots and Dixons

EDITORIAL

Stop Press Number 4 is here! Despite the attraction of the sun kissed beaches of South Wales at the time of compilation, our latest edition comes packed with programs and articles ready for the onset of darker nights and longer Dragon sessions

Many young readers will be starting computer studies in their new school year With a Dragon at home they will have a great opportunity to continue their classroom experiences at leisure Why not teach Mum and Dad to program? After all there's nothing like teaching for helping you to learn!

Stop Press, after four issues, has certainly developed a character of its own, quite different from other computing periodicals We have been encouraged by your letters to believe that its main theme of providing programming material in e helpful way is the right one and long may this continue

Inside this issue there is the, usual Machine Code Corner which carries on from the tast issue to look at some of the techniques involved with moving graphics The Young Users Page concentrates on LOOPS and CIRCLES

Our first competition (Draw a Dragon Logo) produced a good crop of responses from our younger programmers including entries from abroad Congratulations to you all for some professionat programs This issue's competition is extended to ell ages, or even to family entries So get your heads together and send us your programs, on cassette please, to the editorial address -

Miss Cathy Hyde
Dragon Data Ltd
Kenfig Industrial Estete
Margam
Port Talbot
SA13 2PE

The editors have had the privilege of using Dragon Data's new disc system prior to its general release and have been thrilled with its impact on the use of Dragon whether it be in writing programs (the disc system has an automatic line numbering feature which is a boon!) or word processing Stop Press or whatever We are sure that the introduction of this system will offer meny exciting possibilities to Dragon owners for use in their work and we hope to explore this aspect in future editions

Nevertheless we must remember that the reliable and fast cassette interface on Dragon has always been one of its many strong points and we are constantly impressed by Dragon's ability to create and meintain program or data files on tepe (See the letters page for an example of a data file)

Dragon owners with young children (eight yeare upwards?) may wish to find a suitable book to introduce them to the art of Dragon Programming Foulsham have now published 'Dragon Magic' (mentioned in a previous Stop Press) Its large size print and amusing cartoons make it an ideat book for the young beginner Once again the editors invite you to write to us with your hints and suggestions for future articles and programs of interest to other reeders

MACHINE CODE CORNER

In the last edition of Stop Press, we explored e few simple methods of moving shapes around the high resolution graphics screen Our mein concern was vertical movement, since that could be achieved by copying values from byte to byte, without getting involved et the ' bit" levei The time has come to grasp the nettle – and look at horizontal movement

But first we need to learn about a very speciei register – the Condition Code Register (CC) This is a 1-byte (8 bit) register, in which each seperate bit hes e job to do in describing the operetion state of the computer. Each bit is either Ø (clear) or 1 (set) The eight bits of the CC register are as follows

E	F	H	I	N	Z	V	C

E-Entire State Flag
F Fast Interrupt Request Mask
H Half Carry Flag
I-Interrupt Request Mask
N-Sign Flag
Z-Zero Flag
V Overflow Flag
C-Carry Flag

When various machine code commends are executed, these flags are frequently cleared or set according to the result of the command. In fact, we heve often used the zero flag (Z), without referring to it by name One effect of the **CMP** command is to set Z if (end only if) the result is 'true ' For example, **CMPY #$ØØF** will set Z if **Y** is equal to **$6ØF** The **BNE** command makes use of this, by causing a branch if Z is clear So if Z is set by the **CMP** command, there will be no branch

Two commands which give the programmer direct access to CC are ANDCC (opcode 1C) and ORCC (opcode 1A) These perform respectively a logical ' AND" and a logicel "inclusive OR" between CC and the number in the operand

For example, to set Z, we need to perform an inclusive OR with binary ØØØØØ1ØØ i e. with #4. This is because an inclusive OR with a Ø results in "no change' (to all except Z) and an inclusive OR with 1 for Z results in 1 whether or not Z wes previously set As a result **ORCC #4** sets Z. Similarly, to clear Z, we need an AND with binery 11111Ø11, i e. with #251 This is because an AND with 1 results in "no change ' whereas an AND with Ø results in Ø So **ANDCC #251** clears Z

Now let's get beck to horizontel movement You will recall thet in **PMODE4** each pixel of the high resolution screen is represented by one bit, which is either 'on' (green or buff) or 'off (black) Eight of these bits are combined in one byte of memory. The diegram below shows how a black segment of 3 pixels wide is moved to the right a pixel at a time

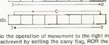
```
byte 1         byte 2
 7654321Ø      7654321Ø

1. 111ØØØ11    11111111
2. 1111ØØØ1    11111111
3  11111ØØØ    11111111
4  111111ØØ    Ø1111111
5  1111111Ø    ØØ111111
6  11111111    ØØØ11111
7  11111111    1ØØØ1111
```

Two bytes ere illustrated, and the bits are labelled from 7 down to Ø, which is the usual convention. We can break down the various operations as follows
(a) a fresh 1' comes in from the left of byte 1,
(b) the bits of byte 1 ere shifted to the right,
(c) bit Ø of byte 1 moves to bit 7 of byte 2
(d) the bits of byte 2 are shifted to the right

If the diagram is read in reverse order (bottom to top) the problem of horizontal movement to the left is seen to be very similar

Two commands which go a long way towards solving the problem are ROR (Rotate Right) and ROL (Rotete Left) These commands rotate the bits *THROUGH THE CARRY FLAG* In other words, ROR has the effect of (a) transferring whatever is in the carry flag to bit 7, (b) rotating bits 7-6-5-4-3-2 1 to 6 5-4 3-2 1-Ø, (c) loading bit Ø into the cerry flag ROL has the reverse effect

A diagram may help



So the operation of movement to the right may be achieved by setting the carry flag, ROR the first byte, ROR the second byte The carry-over between bytes is taken care of autometically by the carry flag.

We shall now use ROR to fire an arrow across the screen First in Basic

```
1Ø PMODE4 1 PCLS1 COLORØ 1 SCREEN1 Ø
2Ø DIMA(2)
3Ø ORAW BM1 5ØRDR2O2LGLE2R12NH2G2
4Ø GET(Ø 5Ø)-(15 54) A
5Ø X$=INKEY$ IFX$=' 'THEN5Ø
6Ø FORI=1TO24Ø PUT(I 5Ø)-(I+15 54) A,PSET NEXT
7Ø GOTO7Ø
```

Note that the DIM statement is necessary for GET/PUT, even though it is not usual to dimension a variable less than 11 The arrow is released by prassing any key Obviously, machine code is called for (which refreshes the parts other languages can't reach) First we must calculate tha address of the bytes making up the arrow. Normally, the graphics scraan will start at hex 6ØØ, but we shall maka the program a little more genaral (to allow for those who may wish to use PMODE4,2 or those who have a Dragon Disk System in operation). The addrass of the top left hand corner of the current grapics screen is contained in memories BA/BB To this value we must add 5Øx32 = 16ØØ since the arrow starts on row 5Ø, and each row is 32 bytes Tha following program does the job

```
                                        Machine code
  1            LDY       $BA            1Ø 9E BA
  2            LEAY      16ØØ Y         31 A9 ØØ ØØ
  3            LDA       #3Ø            8Ø 1E
  4            STA       $7FFF          B7
                                        7FFF
  5  LOOP1     LOA       #8             86 Ø8
  6  LOOP 2    LEAX      ,Y             3Ø A4
  7            LDB       #$15           C8 15
  8  LOOP3     ORCC      #1             1A Ø1
  9            ROR       X              66 84
 1Ø            ROR       1,X            66 Ø1
 11            ROR       2 X            66 Ø2
 12            LEAX      32 X           3Ø 88 2Ø
 13            DECB                     5A
 14            BNE       LOOP3          26 F2
 15            DECA                     4A
 16            BNE       LOOP2          26 EB
 17            LEAY      1,Y            31 21
 18            DEC       $7FFF          7A 7F FF
 19            BNE       LOOP1          26 E2
 2Ø            RTS                      39
```

Lines 1 and 2 give Y the address of tha top-laft byta of the arrow Lines 3 and 4 store 3Ø in **$7FFF** (to kaep a check on the number of times Y is incremented in line 17) A and B are usad for counting – they are decramented until they reach zero Lina 8 sets tha carry flag, so that the first ROR uses a 1 Note that after a DEC command tha zero flag (Z) is set if the result is Ø This is then checked by the BNE command

Tha code may be POKEd in using

```
1Ø OATA 1Ø 9E BA 31 A9 ØØ ØØ 66 1E B7 7F FF 86 Ø8
2Ø OATA 3Ø A4 C8 15 1A Ø1 66 84 66 Ø1 66 Ø2 3Ø 88 2Ø
25 OATA 5A 26 F2 4A 26 EB 31,21 7A 7F FF 26 E2 39
3Ø FORI = ØTO42 REAOX$ POKEI + 32ØØØ VAL(" &H" + X$) NEXT
```

The required Basic program is then

```
1Ø PMOOE4 1 PCLS1 COLORØ 1 SCREEN1 Ø
2Ø ORAW" BM1 5ØRDR2O2LGLE2R12NH2G2"
3Ø X$ = INKEY$ IFX$ = " "THEN3Ø
4Ø EXEC32ØØØ
5Ø GOTO5Ø
```

A much more realistic arrow!

ISLANDS

Can you staar your craft through the islands to the red jetty on the right of the scraan? Unfortunately it is rather foggy so you hava only occasional glimpses of the islands The higher your score the rarer the glimpses! If your craft touches the bottom of the shallow water round an island it sticks there a momant before you can release it If you really run aground you must start back at the beginning Aim for the fastest time You steer your craft with the arrow keys and stop it dead with the spacebar

```
5ØØ FOR I = Ø TO 1 JJ = JOYSTK(I) 32 IF ABS(JJ) < 2Ø THEN 55Ø
```

Now insert NEXT into line 55Ø making it

```
55Ø NEXT X = O(Ø) Y = O(1)*3/4 RETURN
```

Bon voyage.

```
  1 REM ISLANOS  MAVIS PEARSON  KEYBOARD VERSION
 1Ø PCLEAR4 L = 2.S$ = CHR$(8) + CHR$(9) + CHR$(94)
      + CHR$(1Ø) + CHR$(32)
 2Ø PMODE1 1 SCREEN1 Ø PCLS3 TIMER = Ø CLS
 3Ø FORI = 35 TO 21ØSTEP 35  R = RNO(12)*15 CIRCLE(I R) 11,2,7
 4Ø COLOR4 LINE(25Ø 6Ø) (255 1ØØ),
      PSET BF PAINT(I,R) 1 2 NEXT
 5Ø D(Ø) = 1Ø O(1) = 13Ø I = Ø
 6Ø PMOOE1 1 SCREEN1 Ø FORK = 1TO15 NEXT
 7Ø PMOOE1 3 SCREEN1 Ø  PCLS3 FORJ = 1 TO8 + LL
      PSET(X Y 3)
 8Ø GOSUB5ØØ PMODE1 1 P = 4-PPOINT(X,Y)
      PMOOE1 3 PSET(X Y,2)
 9Ø SOUND 1Ø + 5Ø*P,1 ON P + 1 GOTO 12Ø 1Ø6 11Ø 5Ø
1ØØ NEXT GOTO6Ø
11Ø FORK = KK TO 5ØØ NEXT GOTO6Ø
12Ø PRINT@235 "WELL DONE"  "TIME TAKEN"
      FIX(TIMER/5Ø) "SECONDS"
13Ø PRINT INPUT" DO YOU WANT ANOTHER GAME Y/N" O$
14Ø IF O$ = "N"THEN ENO ELSE LL = LL + 1 GOTO2Ø
5ØØ K$ = INKEY$ IFK$ = " "THEN53Ø
51Ø IJ = INSTR(1 S$ K$) IF IJ = 5 THEN JJ = Ø RETURN
52Ø I = (SGN(IJ 2 5) + 1)/2 JJ - IJ-2*I 1 5
53Ø IFO(I) > L THEN O(I) = D(I) + SGN(JJ)*L ELSE D(I) = L + 1
54Ø IF D(I) < 256-L THEN O(I) = D(I) + SGN(JJ)*L ELSE
      D(I) = 256-L
55Ø X = D(Ø) Y = D(1)*3/4 RETURN
```

GETTING INTO LOOPS AND CIRCLES

Do you enjoy making designs with e pair of compasses? I do But I find the point elways slips and somehow the ends don't tie up as well as they should Well your Dragon enjoys drawing circles too There is a special command for circles which mey seem a bit daunting a1 first because it allows you to specify so many things about the circle However the command works as long as you specify three things (it gives the others default settings) Obviously Dragon needs 1o know where you want the centre of your circle to be and whet the radius is So CIRCLE(1ØØ 8Ø),5Ø is the command which tells the machine to draw a circle centred at screen position 1ØØ,8Ø and with radius 5Ø To see this you need a little program -

```
1Ø PCLEAR4 PMODE3 SCREEN1 1 PCLS
4Ø CIRCLE(1ØØ 8Ø) 5Ø
9Ø GOTO 9Ø
RUN
```

Line 1Ø sets up the graphics screen Line 4Ø draws the circle and line 9Ø keeps the program showing the graphics screen Use the BREAK key to stop the program

A circle is a beautiful shape but lots of circles make exciting patterns Let's see how to make them First we'll put the circle command in a subroutine

```
1ØØ CIRCLE(X Y),R C HW START FINISH RETURN
```

This will save a lot of typing and we have all the perameters to play with This means that we must give them values We'll use the default values the machine would give for C,HW,START and FINISH

```
2Ø X=125 Y=1ØØ R=5Ø C=4 HW=1 START=Ø FINISH=1
```

We'll use loops to change the values sterting with the X value Add these lines to your program

```
4Ø FOR X-1Ø TO 24Ø STEP 5
7Ø GOSUB1ØØ
8Ø NEXT
```

Line 4Ø moves the centre of the circle across the screen Line 7Ø calls the subroutine which draws the circle and line 8Ø makes the program go back through the loop until X has a value above 24Ø

Your whole program should now look like thie -

```
1Ø PCLEAR4 PMODE3 SCREEN1 1 PCLS
2Ø X=125 Y=1ØØ R=5Ø C=4 HW=1 START=Ø FINISH=1
4Ø FOR X=1Ø TO 24Ø STEP 5
7Ø GOSUB1ØØ
8Ø NEXT
9Ø GOTO9Ø
1ØØ CIRCLE(X,Y) R,C HW START FINISH RETURN
```

RUN this

To meke the circle drop all we need to do is to type a different line 4Ø

```
4Ø FOR Y=1Ø TO 17Ø STEP 2Ø
```

Now change the 2Ø to 2 end see what happens Of course you can edd more loops and change both X and Y together

Whet ebout chenging 1he radius? Type in this new line 4Ø -

```
4Ø FOR R=1Ø TO 17Ø STEP 7Ø
```

RUN the program and then change line 4Ø again -

```
4Ø FOR R-1Ø TO 17Ø STEP 3
```

Notice that when 1he circle touches the boundary of the screen it is flattened One way to drew a boundary round a screen is to drew a circle with a large radius

The pre1ty pattern above hes an added pattern caueed by interference pa1terns on the screen We cen add colour 1o make i1 more interesting There ere four colours to chose from though, of course one is the background colour Add line 5Ø 1o your program It changes the velue of C from 1 up to 4 and then sets it back to 1 again So each time the machine goes through the loop the colour chenges

```
5Ø IF C<4 THEN C=C+1 ELSE C=1
```

RUN your program now Of course you cen't see the circle drawn in the background colour!

The next parameter is HW or HEIGHT TO WIDTH RATIO It pulls the circle out into an ellipse (like en egg or rugby ball) if HW is less than 1 the circle is flattened If HW is from 1 to 255 the circle is pulled up and down Let's try it with a new line 4Ø

```
40 FOR HW= 01 TO 4 STEP 03
```

When you RUN this you will see a pulsating blob It throbs because every fourth ellipse is drawn in the background colour and over writes some of the previous ellipse. A small alteration in line 5Ø removes this Just change the 1 at the end to 2 Now the background colour is not used

The START and FINISH parameters allow us to draw incomplete circles and ellipses the START and FINISH values go from Ø (3 o'clock) to 1 (clockwise all the way to 3 o'clock again) In fact we can go round nearly four times using values up to nearly 4 We need yet more new line 4Øs and we must clear the screen each time

```
40 FOR START =0 TD 1 STEP 02 PCLS
```

Run the program now and then again with this next line

```
40 FOR FINISH=0 TO 1 STEP 02 PCLS
```

Now it we draw just a little bit of the circle (an arc) each time and increase the radius we should see a spiral We need line 6Ø which increases START from Ø to 1 and back to Ø again and again and keeps FINISH a little bit ahead of START Of course we need to change line 4Ø!

```
40 FOR R=0 TO 80
60 START=R/10 ST=ST-INT(ST) FINISH=START+ 3
```

Rather pretty isn't it? Notice that START and ST are the same as far as Dragon is concerned It uses only the first two letters to identify the variable

Now back to those compass drawings They look best in PMODE4 so change line 1Ø, and change the colour in line 2Ø to C=1 For those of you who know some trigonometry, we need a value for π and can use arctan to obtain one Tan(π/4) = 1 so π = 4*arctan(1) Add or overwrite these lines in your program -

```
30 PI=4*ATN(1) R1=20
40 FOR T=0 TD 360 STEP 15
50 TH=2*PI*T/360
60 X=100+R1*SIN(TH) Y=100+R1*CDS(TH)
```

I hope you like doughnuts! If you prefer dahlias alter the value of R1 in line 3Ø to R1=R

DUCKS

Here is another contribution from Gareth Rowlands You start off with 32 ducks 96 bullets and 75ØØ time units The space bar is your "trigger" If you get them all, you are given another 32 ducks, but not so many bullets and not so much time (your score continues to mount) The number of bullets is controlled by BU and BB in line 31Ø The time limit is controlled by T in the same line

```
  1 REM DUCKS
  2 REM GARETH ROWLANDS, 1983
 10 CLS PCLEAR4 DEF FNA(X)=X-((X=0)*32)-1
 20 GOSUB 310 SOUND 175 1 SOUND175 1
 30 GDSUB 300
 40 GOSUB 340
 50 E=T-TIMER PRINT@11 "TIME"E, X=FNA(X) GDSUB 320
 60 IF INKEY$><" " AND E>0 THEN50
 70 BU=BU-1 IF E <1 THEN 200
 80 SOUND100 1 PRINT@21 "BULLETS" BU
 90 FDR I=1311+G TO 1183+G STEP -64 X=FNA(X)
100 GOSUB320 PDKEI+64 143 POKEI 103 NEXT
110 X=FNA(X) GOSUB 320 POKEI+64 143
120 IF PEEK(I)< >102 THEN PDKEI 103 IF BU<0
    THEN 200 ELSE 50
130 POKEI 164 SCREEN0 1 SOUND200 1
140 I=I-1120 IF X< =I THEN Z=I-X ELSE Z=32 X+I
150 IF Z=0 THEN L$="" ELSE L$=LEFT$(D$ Z)
160 IF Z=32 THEN R$="" ELSE R$=RIGHT$(D$ 31-Z)
170 DK=DK+1 D$=L$+" "+R$ SC=SC+1
    PRINT@0 "SCDRE"SC
180 IF DK=32 THEN 250
190 IF BU>0 THEN 40
200 PRINT@0 "SCDRE"SC
    PRINT@11 "TIME"E PRINT@21 "BULLETS" BU
210 PRINT@160 "WANT ANOTHER GD?",
220 SDUND125 2 SOUND100 2 SOUND60,6
230 A$=INKEY$ IF A$"" THEN 230
240 IF A$="N" THEN CLS END ELSE RUN
250 PLAY "V20T10L3D3CEGO4V30L1C"
260 BB=BB-10 IF BB<34 THEN BB=34
270 T=T-500 IF T<2500 THEN
    T=2500
280 SC=SC+BU BU=BB
290 GOTD 30
300 D$=STRING$(32 30) DK=0 X=32 CLS1
    TIMER=0 RETURN
310 SC=0 BB=90 PR=90 BU=90 T=7500
    RETURN
320 IF X=0 THEN PRINT@PR,D$ RETURN
330 PRINT@PR RIGHT$(D$ X) LEFT$
    (D$ 32-X) RETURN
340 PDKEG+1503 143 POKEG+1471,143 POKEG
    +1439 143 POKEG+1407 143
350 G=RND(32)
360 POKEG+1503 175 POKEG+1471 175 POKEG
    +1439 175 POKEG+1407 175
370 RETURN
```

LETTERS FROM READERS

Readers of STOP PRESS of all ages have aant In letters containing programs, tips and queries All will ba answered in the fullnase of time, end we axpress our thanks to all writers, whether or not their contribution Is included

Two lettars received concerned printera One from D W Abel suggasting that a helpful article on printars would not come amiee confirmed our feelings and has already led to the preparation of an article for the next issue The other letter came from R.R Computer Services, 2a Frithwood Avenue, Northwood, HA6 3LX offering listings of Dragon Programs (cost £2) on raceipt of the program on cassetta.

R J Lentle provided us with a program called INKBLOT which produces "Inkblote" similar to those used In the famous Rorschad Inkblot Tast This is a davice favoured by psychiatrists who ask thair patients to talk about tha shapes Psychologists on the other hand are not impressed but, despite this, we think the program Is of intarest Perhaps our correspondent has programmed Dragon to interpret patients comments!

```
  1 REM INKBLOT  R J LENTLE GOOALMING
 10 PMOOE3 1 SCREEN 1 0 PCLS
 20 DIM R(60)
 30 H =(RNO(30)+30)/100
 40 CIRCLE(123 96) 20 3 H
 50 PAINT(123 96),3 3
 60 GET(100 70)-(150 121) R G
 70 PCLS E= RNO(10) + 29
 80 FOR X=1 TO E
 90 B = RNO(128) A - RND(192)
100 PUT(A B)-(A + 50 B + 41) R OR
110 NEXTX
120 SOUNO 200 4 SOUNO100 4 SOUNO10 6
130 GOTO130
```

Mr T Isaac sende his program JOY-DOODLE which uses tha right joystick On firing the button a line is drawn from the previous point to the current point Pressing "C" re-runs the program clearing the screen. Other possibilities will suggest themselves to interested readers – using the other joystick for further Information such as a centre for a PAINT command Alternatively lines circles and boxes could ba drawn by using other pressad keys

```
  1 REM JOY DOODLE T ISAAC WIGSTON
 10 PMODE4 1 SCREEN1 0 PCLS
 20 PSET(128 96 1)
 30 A = JOYSTK(0) B = JOYSTK(1)
 40 X = FIX(A*4 0) Y = FIX(B*3 5)
 50 PSET(X Y 1)
 60 P = PEEK(65280)
 70 IF INKEY$ = "C" THEN RUN
 80 IF P = 126 OR P = 254 THEN 90 ELSE PRESET(X,Y) GOTO20
 90 LINE (X Y) PSET
100 GOTO 30
```

Mrs Pam D'Arcy Is a keen QUEST player but hae not managed to storm Morlock's castle yet. To continue the same adventure at a future time she offers the following procadure to dump variable values to tape –

Load Quest and type In the following lines –

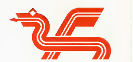
```
   10 CLEAR432 OIMO$(9) O$(17) D(28 4) O(17 6) CLS
 1000 GOSUB10010 PRINT@PL "WHAT NOW SIRE?",
       A$= "IUNSEWOR"
 1010 IF O=7 THEN25000 ELSE IF O =B
      THEN 26000  REM OUIT
      WITH SAVE OR LOAD NEW FILE RESP EC
25000 REM SAVE OATA FILE
25010 G$= "SAVEO"
25020 GOSUB 27000 REM GET FILE NAME
25030 OPEN "O" #-1 G$
25040 PRINT #-1 A AL A$ B BC
      BI,BM BO BT BW,BX B$ C CH DF O$
25050 FOR G=1 TO 9 PRINT #-1,D$(G) NEXT G
25060 FOR G=1 TO 28 PRINT #-1 D(G 1) O(G,2) O
      (G 3) O(G,4) NEXTG
25070 PRINT #- 1,
      EG EX E$ F FA F1 HB HC,H$ I
      IW J K L M MC MF M$ N OC,OL
25080 FOR G=1 TO 17 PRINT #-1 O$(G) O(G,1)
      O(G 2) O(G 3) O(G 4) O(G 5) O(G 6) NEXTG
25090 PRINT #-1 P PL O OT R$ SB,
      SF SL SP ST,SU SWF S1 TW V,VF,
      WT WV W$ X Y Y$,Z,ZL ZZ
25100 CLOSE #-1 PRINT PRINT "OATAFILE ",G$," SAVED" END
26000 REM LOAO OATA FILE
26010 G$= "LOADED "
26020 GOSUB27000
26030 OPEN "I" #-1 G$
26040 INPUT #-1 A AL,A$ B BC
      BI BM BO BT 6W BX B$ C CH OF O$
26050 FOR G=1 TO 9 INPUT # 1 O$(G) NEXTG
26060 FOR G=1 TO 28 INPUT #-1 O(G 1) D(G,2),O(G,3),O(G 4)
      NEXTG
 26070   INPUT #  1 EG,EX E$ F FA,F1 HB HC,H$,I IW,J
         K L M MC MF M$ N OC OL
26080 FOR G=1 TO 17 INPUT # 1,O$(G) O(G,1),
      O(G 2) O(G 3) O(G 4) O(G 5) O(G 6) NEXTG
26090 INPUT #-1 P PL O OT R$ SB SF
      SL SP ST SU SWF S1 TW V VF WT,WV
      W$ X Y Y$ Z,ZL ZZ
26100 CLOSE # 1
26110 GOSUB 9000 GOTO1000 REM
      PRINT CURRENT SCREEN & CONTINUE
27000 REM SAVE/LOAO PRINT & INPUT FILENAME
27010 CLS
27020 PRINT PRINT "GET TAPE AND RECOROER REAOY FOR
      FILE TO BE ", G$ "ENTER NAME TO FILE TO BE " G$
27030 INPUT G$
27040 RETURN
```

Mrs D'Arcy aeeuras us thet it works! Wa added one screen comment and although we have not givan it a full test we think Quast players will find it very ueaful

DRAW A DRAGON LOGO COMPETITION

The editors were delighted with the response to the first competition, with entries from boys and girls both in the U.K. and abroad. After considerable deliberation, Graham Wademan's entry was chosen as the winner of this competition. His logo was excellent and the whole program was professionally packaged. (See for yourself by typing in the listing below.)

All the logos had their own personalities. There were fat, thin, friendly and fiery dragons! One or two entries for the competition did not draw a dragon logo at all and therefore had to be excluded. Nevertheless we appreciate your programs and we print below one such entry that is skilfully constructed and represents a dragon flying! Thank you Bobby Patel for this program.



```
10 PCLS PMODE1 SCREEN1 0
20 DRAW "BM40.50 R24 BM+10 0 R10"
30 CIRCLE(68 40) 4,2 PAINT(68 40) 3 2
40 DRAW "F9D11 G0R100F10D13G13L9E13U13H3L110U5E10
   U0H3L35H50M170 113 E15U12L115U9E10
   U7L45U6G8F6U8BM+10 +5R23G10D15R114D7G16L9E13U3
   L192H8U18E5L9H48M40 50R24F5E5
50 PAINT(44 52) 8 8 PAINT(74 54) 8 8 PAINT(64 60) 8,8
   PAINT(32 60) 8 8
60 DRAW "BM70 113 R9E12L9G12R9 BM+3 0E12 BM +5.0
   G12R9E13L9
70 PAINT(82 104) 8 8 PAINT(96 104) 8 8
80 DRAW" BM +30,-28 E20R25E15L34G36R9L9BM-4.0 E39 R37 BM
   +5,-4 E7L62G50R8E43R42
90 PAINT(100 44) 8 8 PAINT(100,25) 8 8
100 DRAW" BM20 10C2R214D115L214U115 BM-3,-3 C3
   R221D121L221U121" PAINT(16 100),2 3
110 DRAW "BM40 150 C3 D3 BM+5,0 D12R2E4U4H4L2BM+
   10 0 D12U5R3F5H5E3U1H3L2 BM+10 +12
   U6R2U6R5D6L5R6D6 BM+4 0
   R7U4L3R3D4L7U12R0D2BM+4 0
   U2D12R0U12L8R8BM+4 0D12U12F12U12BM+
   60,0 D12R8BM+4,0U12R8D12L8R8 BM+4 0"
120 DRAW "R6U4L3R3D4L6U12R6D2U2BM+4 0
   D12R8U12L8R8BM+3,0U3 "
130 DRAW" BM+0 -5C4L183D26R183U26"
140 DRAW "BM+5 -5 C1 L192D35R192U35"
150 GOTO150
```

30

```
10 CLS CLEAR1000
20 A$="" SP$=STRING$(27 32)
30 FORN=1TO4 FORJ=1TO5 READCR
40 A$=A$+CHR$(CR) NEXTJ
50 A$=A$+SP$ NEXTN
60 FORN=1TO3 B$=B$+STRING$(5 143)
   +STRING$(27 32) NEXT B$=B$+STRING$(5 143)
70 FORX=12TO332STEP1
80 PRINT@X A$ FORJ=1TO20 NEXT PRINT@X B$
90 NEXT
100 END
110 DATA 143 143 128 128 128
120 DATA 143 143 128 141 143
130 DATA 132 143 128 128 143
140 DATA 128 128 128 128 143
```

SOLUTION TO DRAGON PUZZLE 2

The numerical solutions to the clues gave the appropriate line numbers for the program statements to play the tune 'Oh when the saints go marching in'. The solutions were in order 1914,20,1492, 13, 1981,1918,1066,18.

COMPETITION

WIN 4 SOFTWARE CASSETTES

WIN 4 SOFTWARE CASSETTES (COMPETITION)

Four ants start from the four corners of the **PMODE**4 screen and travel in such a way that each ant always travels towards the nearest ant in the clockwise direction. All movement is at the same uniform speed. Write a program to demonstrate the nature of the paths taken by the ants.

The competition is open to all ages and the best solutions in each age group will receive the choice of free Dragon software. Send your entries to the editorial address on a cassette together with your name and address and age (if under 16) not later than September 30th.

DRAGON the teacher

Educational software is now the 'in' thing with the boom in home computers Commercial educational software is usually sophisticated and consequently time-consuming to write. Nevertheless small programs designed to do a particular task can be fun to write with the advantage that they may be tailor made to your own specific needs

Take your daughter or son at secondary school who has a language vocabulary to learn each week The program following is simple and to the point, allowing the user to input a vocabulary in two languages and then answer randomly chosen questions

```
 1 REM LANGUAGE TESTER A M SYKES
10 INPUT' HOW MANY WORDS  N DIM A$(2 50)
20 INPUT FIRST LANGUAGE  B$(1) INPUT' SECDND
   LANGUAGE ' B$(2)
30 CLS FOR I=1 TO N FOR J=1 TO 2 PRINT' ENTER WDRD
   I 'IN  B$(J) INPUT A$(J I) NEXTJ t
40 I=RND(N) J=RND(2) IF J=2 THEN K=1 ELSE K=2
50 CLS PRINT 'WHAT IS THE ' B$(J),  WDRD FDR  A$(K,I)
60 INPUT AN$ IF AN$=A$(J I) THEN PRINT
   'WELL DDNE  GOSUB70 GOTO40  ELSE PRINT
   'NO — TRY AGAIN' GOSUB70 GDTO50
70 FDR D=1 TD 400 NEXT RETURN
```

Or consider building up a vocabulary of English words and their opposites. This can be done with DATA statements so that information can be recorded on tape and extended as and when required

To have this tacility, tha program is constructed using a succession ot X s at the end ot the DATA statemants This serves not only to provide a stop to tha READing ot tha data, but also to leave space on the tape recording tor subsequent recordings ot an extended version

```
 1 REM OPPOSITES A M SYKES
10 C=1 DIM A$(2 50)
20 FOR I-1 TD 2 READA$(I C) IF LEFT$(A$(I C) 3)= 'XXX'
   THEN 30 ELSE NEXT I C=C+1 GOTO20
30 I-RND(C) J=RND(2) IF I=II THEN 30 ELSE II=I
40 [illegible]
50 CLS:PRINT'WHAT IS THE OPPOSITE OF ";A$(J,I)
60 INPUT AN$ IF AN$-A$(K I) THEN PRINT 'WELL
   DDNE  GDSUB70 GDTD30 ELSE PRINT "TRY
   AGAIN  GOSUB70 GOTO50
70 FOR D=1 TO 400 NEXT RETURN
80 DATA ASLEEP AWAKE ADD SUBTRACT
   LIVELY DULL SOBER DRUNK FALL RISE LARGE SMALL
   FAST,SLOW,UGLY BEAUTIFUL NEAR,FAR HIGH
   LDW,SILLY,SENSIBLE,LIGHT DARK XXXXXXXX
```

These two programs are useful but rather plain They also suffer (as far as young children are concerned) in damanding a typed response The last program overcomes this problem by presenting a numbered list With the addition of movement and sound it also becomes more interesting to the user Try it and see if you can correctly identify the second member of each pair (Apologies to Dragon owners in Huddersfield for the repetition!)

```
  1 REM PAIRS A M SYKES
 10 DIMA(15),A$(2 15) F(16) FOR I=1 TO 15 A(I)=I NEXT
 20 FOR I =15 TO1STEP-1 C=RND(I) W=A(C)
    READA$(1 I) A$(2 I) A(C)-A(I) A(I)=W NEXTI
 30 FDR I=1TD15 PRINT@32*I-32 A$(1,I)+  AND  ' NEXTI
 40 GDSUB80 FDR I=1TD 10
 50 PRINT@32*I 19 "" INPUTN IF A(N)<>I
    THEN SDUND10 8 GDSUB80 GDTD50
 60 F(N)=1 GDSUB80 GDSUB100 NEXTI
 70 PLAY L6BL3O4C  STOP
 80 FOR  J-1TO15 IFF(J)=1THEN  NEXTJ  ELSEPRINT@32*J-
    11 J A$(2 A(J)) NEXTJ
 90 RETURN
100 PLAY T5O4CO3L10GF#GL4G#G ' FOR K=1TO9 PR1NT@32*
    N 9 K A$(2 I)+STRING$(K+2 32) NEXTK
110 IF N>I THEN ST=-1 ELSE ST-1
120 FDR K=N TD I STEP ST PRINT@32*K 19 A$(2 I) FOR L=1
    TO 50 NEXTL PRINT@32*K 19 STRING$(7,32)
    NEXTK PRINT@32*I-19 A$(2,I), FOR M-1TO I
    PRINT@32*M-19 A$(2,M)
    NEXTM RETURN
130 DATA BOOTS SHDES SDAP WATER KNIFE FDRK
    NEEDLE THREAD BISCUITS TEA BACON EGG FISH
    CHIPS LAND SEA CLOAK DAGGER HARD FAST
    [illegible]
    PEEK,POKE,KING,QUEEN,SALT,PEPPER
```

More new titles . . .

Watch out tor the following new titles in the official Dragon software list They will be appearing in your shops soon

Viking
Monsters and Magic
Words Words Words
Lats Count

Bloc Head
Narbie Force
Adventure Trilogy
Beyond the Cimeeon Moon

Forth
Bridge
Moon Hopper
Jumpjet

Lunar Rover Patrol
Cosmic Clones
El Bandito
Star Fighter
Super Dragon Writer 11

Published by Welbeck Public Relations 2 Endell Street London WC2 Printed by Orydans Printers Ltd., Brent Crescent North Circular Road London NW10